

Chương 4 : PHP & MySQL

Giảng Viên: ThS. Vũ Minh Sang

Nội dung



- Giới thiệu các cách kết nối PHP với MySQL
- Các bước thao tác với CSDL
- Các câu lệnh truy vấn CSDL
- Font chữ khi thao tác với MySQL



Giới thiệu



- Hiện nay, có 2 phương thức để kết nối PHP với MySQL: MySQLi và PDO.
- Mặc định khi cài PHP thì cả 2 thư viện MySQLi và PDO đã được tích hợp.
- Lưu ý: thư viện MySQL đã không còn được sử dụng từ phiên bản PHP7 trở lên.





Các bước thao tác kết nối CSDL trong PHP

- Kết nối với CSDL
- Chọn CSDL
- Xây dựng câu truy vấn dữ liệu
- Thực thi câu truy vấn
- Xử lý kết quả trả về từ câu truy vấn SELECT, INSERT, UPDATE, DELETE
- Ngắt kết nối với CSDL



Sử dụng MySQLi



Giới thiệu



- MySQLi (MySQL Improved):
 - Là dạng kết nối thường được sử dụng
 - Được tích hợp sẵn trong PHP từ phiên bản 5.0
 - Tận dụng được các tính năng mới của MySQL phiên bản 4.1.3 trở lên.
 - Hỗ trợ giao tiếp theo hướng thủ tục (procedural) và hướng đối tượng (object-oriented).
 - Xem dữ liệu như là bộ các đối tượng với functions, người dùng thêm xóa dữ liệu dễ dàng.
 - Hỗ trợ dạng **prepared statements**, tăng cường khả năng chống SQL Injection
 - Nhược điểm: không làm việc được với các hệ quản trị CSDL khác, ngoài MySQL





Sử dụng MySQLi (dạng procedural)

- Sử dụng các hàm của MySQLi để thao tác trên CSDL
 - mysqli_connect
 - mysqli_select_db
 - mysqli_query
 - mysqli_num_rows
 - mysqli_fetch_array
 - mysqli_close

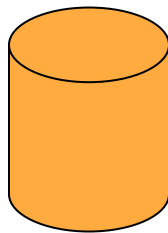


Tạo kết nối CSDL



PHP Pages

Connection



Database

- Kết nối MySQL và chọn CSDL

```
resource mysqli_connect(string host, string LoginName, string password)  
int mysqli_select_db(string dbName, resource connection)
```

Hoặc

```
resource mysqli_connect(string host, string LoginName, string password, string dbName)
```




Tạo kết nối CSDL

- Cách 1

<?php

```
$SERVER = "localhost";
$USERNAME = "root";
$PASSWORD = "";
$DBNAME = "qlsv";
$conn = mysqli_connect($SERVER, $USERNAME, $PASSWORD);
if ( !$conn ) {
    //Không kết nối được, thoát ra và báo lỗi
    die("Không kết nối được vào MySQL server");
} //end if
echo "Kết nối thành công";
//chọn CSDL để làm việc
mysqli_select_db($conn, $DBNAME);
//đóng kết nối
mysqli_close($conn);
```



Tạo kết nối CSDL



• Cách 1

<?php

```
$SERVER = "localhost";  
$USERNAME = "root";  
$PASSWORD = "";  
$DBNAME = "qlsv";  
$conn = mysqli_connect($SERVER, $USERNAME, $PASSWORD);  
    //Không kết nối được, thoát ra và báo lỗi  
    Or die("Không kết nối được vào MySQL server" .mysqli_connect_error());  
  
echo "Kết nối thành công";  
    //chọn CSDL để làm việc  
mysqli_select_db($conn, $DBNAME);  
    //đóng kết nối  
mysqli_close($conn);
```

?>



Tạo kết nối CSDL



• Cách 2

```
<?php
$SERVER = "localhost";
$USERNAME = "root";
$PASSWORD = "";
$DBNAME = "qlsv";
$conn= mysqli_connect($SERVER, $USERNAME, $PASSWORD,$DBNAME) ;
if ( !$conn ) {
    //Không kết nối được, thoát ra và báo lỗi
    die("Không kết nối được vào MySQL server");
} //end if
echo "Kết nối thành công";
//đóng kết nối
mysqli_close($conn);
```

?>



Tạo kết nối CSDL



• Cách 2

```
<?php
$SERVER = "localhost";
$USERNAME = "root";
$PASSWORD = "";
$DBNAME = "qlsv";
$conn= mysqli_connect($SERVER, $USERNAME, $PASSWORD,$DBNAME);
    //Không kết nối được, thoát ra và báo lỗi
    Or die("Không kết nối được vào MySQL server");
echo "Kết nối thành công";
//đóng kết nối
mysqli_close($conn);
?>
```



Xây dựng câu truy vấn



SQL	Ví dụ
INSERT	<code>strSQL = "INSERT INTO Users (UserName, Password) VALUES ('admin', '123')"</code>
DELETE	<code>strSQL = "DELETE FROM Users WHERE UserName = 'lvminh'"</code>
UPDATE	<code>strSQL = "UPDATE Users SET Password = 'abc' WHERE UserName = 'admin'"</code>
SELECT	<code>strSQL = "SELECT * FROM Users WHERE UserName = 'admin'"</code>

Thực thi câu truy vấn

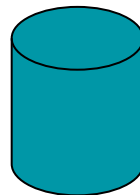


PHP Pages

011010011

Connection

011010011



Database

```
resource mysqli_query(resource connection, string strSQL)
```

```
<?php
    $strSQL = "INSERT INTO Users (UserName, Password) VALUES ('admin','123')";
    mysqli_query($conn, $strSQL);
?>
```

```
<?php
    $strSQL = "SELECT * FROM Users";
    $result = mysqli_query($conn, $strSQL);
?>
```



Thực thi câu truy vấn



- Giá trị trả về của hàm **mysqli_query**

- Không trả về giá trị với câu lệnh sql **CREATE DATABASE** và **CREATE TABLE**

```
$strSQL = "Create table User (id int AUTO_INCREMENT primary key,  
Username varchar(20), Password varchar(30))";
```

```
mysqli_query($conn, $strSQL);
```

- Đối với câu lệnh **SELECT** trả về kết quả thu được

```
$result = mysqli_query($conn, $strSQL);
```

- Đối với câu lệnh **INSERT**, **UPDATE**, **DELETE** trả về giá trị 1 (true), 0 (false)



Xử lý kết quả trả về



- Hiển thị dữ liệu trả về

- Sử dụng hàm

- `$row = mysqli_fetch_array($result)`

- `$row = mysqli_fetch_assoc($result)`

- `$row = mysqli_fetch_row($result)`

- Số lượng FIELD, ROW

- `$num = mysqli_num_fields($result);`

- `$num = mysqli_num_rows($result);`

- Truy cập đến từng FIELD

- `echo $row["UserName"];`

- Hoặc

- `echo $row[0];`

UserName	Password
A	AA
B	BB
C	CC
D	DD
...	...
X	XX
Y	YY
Z	ZZ

cursor ←

Xử lý kết quả trả về



- `$rowArray = mysqli_fetch_array($result, MYSQLI_ASSOC | MYSQLI_NUM | MYSQLI_BOTH)`
`while ($rowArray = mysqli_fetch_array($result)) {`
 `echo $rowArray[0] . $rowArray["UserName"];`
`}`
- `$rowAssoc = mysqli_fetch_assoc($result)`
`while ($rowAssoc = mysqli_fetch_assoc($result)) {`
 `echo $rowAssoc["UserName"];`
`}`
- `$row = mysqli_fetch_row($result)`
`while ($row = mysqli_fetch_row($result)) {`
 `echo $row[1];`
`}`



```
<?php
```

```
$server = "localhost";
$username = "root";
$password = "";
$dbName = "qlsv";
$conn = mysqli_connect($server, $username, $password, $dbName)
    Or die("<p> Không thể connect</p>");
mysqli_set_charset($conn,"utf8");
$strSQL = "select * from User";
$result = mysqli_query($conn, $strSQL)
    Or die("<p>Không thể thực thi câu truy vấn.</p>");
echo "<table border =1 >";
echo "<tr><th>Username</th><th>Password</th></tr>";
while ($row = mysqli_fetch_row($result)){
    echo "<tr>";
    echo "<td>" . $row[0] . "</td>";
    echo "<td>" . $row[1] . "</td>";
    echo "</tr>";
};
echo "</table>";
//Đóng kết nối
mysqli_close($conn);
```

```
?>
```

Showing rows 0 - 3 (4 total, Query took 0.0274 sec)

```
SELECT *
FROM `user`
LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers
Sort by key: None
+ Options

	UserName	PassWord
<input type="checkbox"/>	Mai Xuân Hùng	0901
<input type="checkbox"/>	Thái Bá Thiện	0902
<input type="checkbox"/>	Nguyễn Hữu Đạt	0903
<input type="checkbox"/>	Nguyễn Văn Phước	0904

user name	password
Mai Xuân Hùng	0901
Thái Bá Thiện	0902
Nguyễn Hữu Đạt	0903
Nguyễn Văn Phước	0904



Ngắt kết nối dữ liệu



- Ngắt kết nối cơ sở dữ liệu

```
mysqli_close($conn) ;
```

- Tự động thực thi khi kết thúc mã lệnh nhờ cơ chế tự động thu rác (garbage collector) – PHP4,5
- Giải phóng vùng nhớ chứa dữ liệu sau khi truy vấn

```
mysqli_free_result($result) ;
```



Tổng hợp các bước thao tác CSDL MySQL



<?php

```
$SERVER = "localhost";
$USERNAME = "root";
$PASSWORD = "";
$DBNAME = "mydata";
$conn = mysqli_connect($SERVER, $USERNAME, $PASSWORD);
if ( !$conn ) {
    //Không kết nối được, thoát ra và báo lỗi
    die("Không kết nối được MySQL server:". mysqli_connect_error());
} //end if
//chọn CSDL để làm việc
mysqli_select_db($conn, $DBNAME)
or die("Không thể chọn được CSDL: ". mysqli_error($conn));
```

?>



Tổng hợp các bước thao tác CSDL MySQL



- Sử dụng câu lệnh `mysqli_query($conn, $sql);`

- Thêm:

```
$strSQL = "Insert into User values('Mai Xuân Hùng', '0901')";  
$result = mysqli_query($conn, $strSQL)  
    Or die("<p>Không thể thực thi câu truy vấn.</p>"  
    . "<p>Error code " . mysqli_errno($conn)  
    . ": " . mysqli_error($conn) . "</p>");  
echo "Thêm thành công";
```

Tổng hợp các bước thao tác CSDL MySQL



- Sửa, Xóa

```
$strSQL = "Update User Set Username = 'Mai Xuân Đến' where Password = '0901')";  
$result = mysqli_query($conn, $strSQL)  
    Or die("<p>Không thể thực thi câu truy vấn.</p>"  
    . "<p>Error code " . mysqli_errno($conn)  
    . ": " . mysqli_error($conn) . "</p>");  
echo "Cập nhật thành công" . mysqli_affected_rows($conn) . "dòng" ;
```



Tổng hợp các bước thao tác CSDL MySQL



- Câu lệnh select

```
$sql = "SELECT * FROM HS";  
$result = mysqli_query($conn, $sql);  
if ( !$result )  
    die("Không thể thực hiện được câu lệnh SQL: ".mysqli_error($conn));  
echo "Số lượng row tìm được: ".mysqli_num_rows($result)."<br>";  
while ( $row = mysqli_fetch_row($result) ) {  
    echo "Ten: ".$row[0];  
    echo "    Ma so: ".$row[1]."<br>";  
}  
//end while  
//nên luôn giải phóng bộ nhớ sau khi lấy hết các row trả về từ câu lệnh SELECT  
mysqli_free_result($result);  
//đóng kết nối  
mysqli_close($conn);
```



Sử dụng Prepared statements



```
$sql = "SELECT * FROM HS where ID=?";
$id = 6;
if ( $stmt = mysqli_prepare($conn, $sql)){
    //Truyền dữ liệu vào tham số
    mysqli_stmt_bind_param($stmt, "i", $id)
    //Thực thi câu truy vấn
    mysqli_stmt_execute($stmt);
    //Gán dữ liệu trả về vào biến $result
    mysqli_stmt_bind_result($stmt, $result);
    //Thực hiện fetch dữ liệu
    mysqli_stmt_fetch($stmt);
    //in ra kết quả trả về
    printf('Result: %s', $result);
    //close mysqli_stmt
    mysqli_stmt_close($stmt);
}
```



Sử dụng Prepared statements



- Tham số ẩn danh trong câu truy vấn dùng ký tự ?
- Hàm `mysqli_stmt_bind_param($stmt, "i", $id)` :
 - Giá trị thứ 2 "i" chính là khai báo kiểu dữ liệu cho các tham số ẩn danh trong câu truy vấn. Các giá trị như sau:
 - i: interger
 - d: double
 - s: string
 - b: blod
 - Nếu nhiều tham số ẩn danh thì sẽ khai báo kiểu cho đủ số tham số ẩn danh. Vd: câu truy vấn có 2 tham số ẩn danh kiểu string: khai báo "ss"
 - Tham số cuối cùng (\$id) là các tham số dạng tham chiếu, không cần phải được khai báo trước khi gọi hàm.

Sử dụng Prepared Statements



• Câu lệnh Insert

```
// Chuẩn bị câu lệnh INSERT
$sql = "INSERT INTO persons (first_name, last_name) VALUES (?, ?)";
if($stmt = mysqli_prepare($conn, $sql)){
    // Liên kết biến với tham số trong câu lệnh đã chuẩn bị
    mysqli_stmt_bind_param($stmt, "ss", $first_name, $last_name);
    // Thiết lập tham số lần 1
    $first_name = "abc";
    $last_name = "xyz";
    // Thực hiện câu lệnh
    mysqli_stmt_execute($stmt);
    // Thiết lập tham số lần 2
    $first_name = "efg";
    $last_name = "vmk";
    // Thực hiện câu lệnh
    mysqli_stmt_execute($stmt);
}
```





Sử dụng MySQLi (dạng object oriented)

- Sử dụng các hàm của đối tượng MySQLi để thao tác trên CSDL

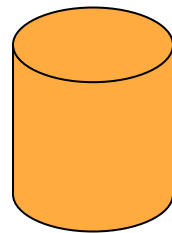


Tạo kết nối CSDL



PHP Pages

Connection



Database

- Khởi tạo đối tượng để kết nối

```
$conn = new mysqli("hostname", "username", "password", "database");
```

- Kiểm tra kết nối

```
if ($conn->connect_error) {  
    die("Kết nối lỗi: " . $conn->connect_error);  
}  
echo "Kết nối thành công!!!";
```



Tạo kết nối CSDL



```
<?php
```

```
$SERVER = "localhost";  
$USERNAME = "root";  
$PASSWORD = "";  
$DBNAME = "qlsv";  
$conn = new mysqli($SERVER, $USERNAME, $PASSWORD, $DBNAME);  
if ($conn->connect_error) {  
    die("Kết nối lỗi: " . $conn->connect_error);  
}  
echo "Kết nối thành công!!!";
```

```
?>
```



Thực thi câu truy vấn Insert



- Sử dụng hàm **query** của đối tượng **mysqli**

```
$strSQL = "Insert into User values('Mai Xuân Hùng', '0901')";  
if ($conn->query($strSQL) === true)  
    echo "Thêm thành công!!!";  
}else echo "Thêm không thành công!!!";
```





Thực thi câu truy vấn Delete

- Sử dụng hàm **query** của đối tượng **mysqli**

```
$strSQL = "Delete from User where Password = '0901'";  
if ($conn->query($strSQL) === true)  
    echo "Xóa thành công!!!";  
}else echo "Xóa không thành công!!!";
```



Thực thi câu truy vấn Update



- Sử dụng hàm **query** của đối tượng **mysqli**

```
$strSQL = "Update User Set Username = 'Mai Xuân Đến' where Password = '0901'";
```

```
if ($conn->query($strSQL) === true)
```

```
    echo "Cập nhật thành công!!!";
```

```
}else echo "Cập nhật không thành công!!!";
```





Thực thi câu truy vấn Select

- Sử dụng hàm **query** của đối tượng **mysqli**

```
$strSQL = "select * from User";
```

```
$result = $conn->query($strSQL)
```



Thực thi câu truy vấn Select



- Hiện thị dữ liệu trả về

- Sử dụng các hàm của biến **\$result**

- \$row = **\$result->fetch_array()**

- \$row = **\$result->fetch_assoc()**

- \$row = **\$result->fetch_row()**

- Số lượng FIELD, ROW

- \$num = **\$result->num_fields()** ;

- \$num = **\$result->num_rows()** ;

- Truy cập đến từng FIELD

- echo **\$row**["UserName"] ;

- Hoặc

- echo **\$row**[0] ;

UserName	Password
A	AA
B	BB
C	CC
D	DD
...	...
X	XX
Y	YY
Z	ZZ

cursor ←

Xử lý kết quả trả về



- `$rowArray = $result->fetch_array(int $resulttype = MYSQLI_ASSOC | MYSQLI_NUM | MYSQLI_BOTH)`
`while ($rowArray = $result->fetch_array()) {`
 `echo $rowArray[0] . $rowArray["UserName"];`
`}`
- `$rowAssoc = $result->fetch_assoc()`
`while ($rowAssoc = $result->fetch_assoc()) {`
 `echo $rowAssoc["UserName"];`
`}`
- `$row = $result->fetch_row()`
`while ($row = $result->fetch_row()) {`
 `echo $row[1];`
`}`



Sử dụng Prepared Statements



```
$sql = "SELECT * FROM HS where ID=?";  
$id = 6;  
$stmt = $conn->prepare($sql)  
//Truyền dữ liệu vào tham số  
$stmt->bind_param("i", $id)  
//Thực thi câu truy vấn  
$stmt->execute();  
//Gán dữ liệu trả về vào biến $result  
$stmt->bind_result($result);  
//Thực hiện fetch dữ liệu  
$stmt->fetch();  
//in ra kết quả trả về  
printf('Result: %s', $result);
```



Ngắt kết nối dữ liệu



- Giải phóng vùng nhớ chứa dữ liệu sau khi truy vấn

```
$result->close () ;
```

- Ngắt kết nối cơ sở dữ liệu

```
$conn->close () ;
```





Sử dụng PDO



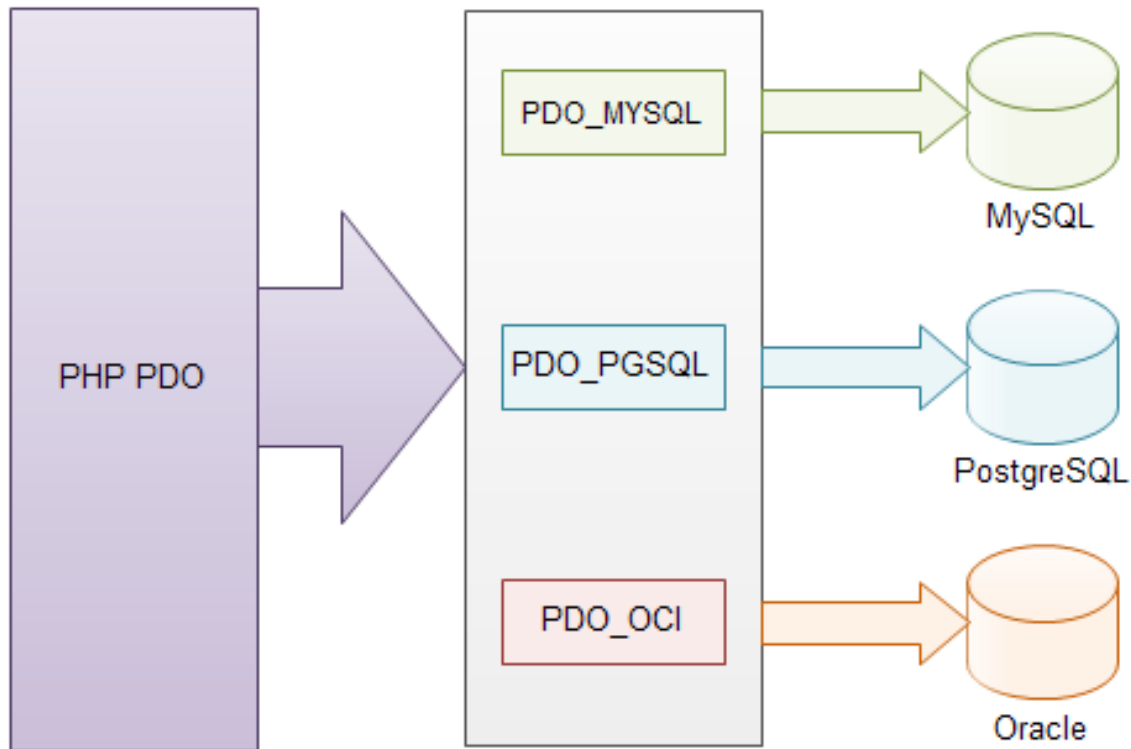
Giới thiệu



- PDO (PHP Data Object):
 - Hỗ trợ 12 loại hệ quản trị CSDL khác nhau.
 - Là dạng database abstraction layer; sử dụng các extension để giao tiếp với các CSDL khác nhau.
 - Mỗi hệ quản trị khác nhau sẽ có các driver khác nhau để chuyển đổi sang câu lệnh SQL tương ứng.
 - Là dạng hướng đối tượng (object oriented)
 - Hỗ trợ dạng **prepared statements** và **name parameters** giúp tăng cường khả năng chống SQL Injection



Giới thiệu



Tạo kết nối CSDL



- Mỗi hệ quản trị CSDL sẽ có phương thức kết nối khác nhau (có loại cần Username, Password, đường dẫn tới Database, Port, có loại không)
- Với MySQL:

```
$conn = new PDO("mysql:host=localhost;dbname=database", 'username', 'password');
```





Tạo kết nối CSDL

- VD:

```
<?php
```

```
    $SERVER = "localhost";
```

```
    $USERNAME = "root";
```

```
    $PASSWORD = "";
```

```
try{  
    $conn = new PDO("mysql:host=$SERVER;dbname=qlsv",$USERNAME,$PASSWORD);  
    // Thiết lập lỗi của PDO cho ngoại lệ  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    echo Kết nối thành công!!!";  
}catch(PDOException $e) {  
    echo "Kết nối thành công!!!". $e->getMessage();  
}  
?>
```



Thực thi câu truy vấn Select



- Khi thực thi câu truy vấn select, PDO trả dữ liệu về theo mảng (array) hoặc đối tượng (object) bằng phương thức **fetch()**.
 - **PDO::FETCH_ASSOC**: Trả về dữ liệu dạng mảng với key là tên thuộc tính trong bảng của câu truy vấn select.
 - **PDO::FETCH_BOTH (default)**: Trả về dữ liệu dạng mảng với key là tên và cả số thứ tự của thuộc tính
 - **PDO::FETCH_BOUND**: Gán giá trị của từng thuộc cho từng biến đã khởi tạo trước đó qua phương thức bindColumn()
 - **PDO::FETCH_CLASS**: Gán giá trị của từng thuộc tính (column) của bảng cho từng thuộc tính (property/attribute) của một lớp Class theo tên column và tên thuộc tính.
 - **PDO::FETCH_INTO**: Gán giá trị của từng column cho từng thuộc tính của một Class Instance (thể hiện của một lớp)

Thực thi câu truy vấn Select



- **PDO::FETCH_LAZY**: Gộp chung PDO::FETCH_BOTH/PDO::FETCH_OBJ
- **PDO::FETCH_NUM**: Trả về dữ liệu dạng mảng với key là số thứ tự của thuộc tính
- **PDO::FETCH_OBJ**: Trả về một Object của stdClass (link is external) với tên thuộc tính của Object là tên của column.
- Cần thiết lập dạng PDO Fetch trước khi sử dụng bằng: **setFetchMode**
- VD: `$stmt->setFetchMode(PDO::FETCH_ASSOC);`
- Các kiểu fetch thường dùng: **FETCH_ASSOC**, **FETCH_CLASS** và **FETCH_OBJ**



Thực thi câu truy vấn Select (FETCH_ASSOC)



```
//Tạo Câu truy vấn và thực thi
$stmt = $conn->query('SELECT * From USER');
//Thiết lập kiểu dữ liệu trả về
$stmt->setFetchMode(PDO::FETCH_ASSOC);
//Hiển thị kết quả
while($row = $stmt->fetch()) {
    echo $row['username'] , '\n';
    echo $row['password'];
}
```





Thực thi câu truy vấn Select (FETCH_OBJ)

- Kết quả trả về là một object của stdClass cho mỗi dòng kết quả

```
//Tạo câu truy vấn và thực thi
```

```
$stmt = $conn->query('SELECT * From USER);
```

```
//Thiết lập kiểu dữ liệu trả về
```

```
$stmt->setFetchMode(PDO::FETCH_OBJ);
```

```
//Hiển thị kết quả
```

```
while($row = $stmt->fetch()) {
```

```
    echo $row->username, '\n';
```

```
    echo $row->password;
```

```
}
```



Thực thi câu truy vấn Select (FETCH_CLASS)



- Kết quả trả về là một object theo class được chỉ định (ánh xạ đối tượng)
- Thuộc tính của class sẽ được gán trước khi hàm khởi tạo của class đó được gọi.
- Nếu **không** có thuộc tính **khớp** với tên cột trong bảng thì tự động **tạo** một thuộc tính public trong class

```
class User {  
    public $username;  
    public $password;  
    public $isAdmin = 'No';  
    function __construct() {  
        if ($this->name == 'Microsoft Window')  
            $this->isAdmin = 'Yes';  
    }  
    public function display(){  
        return $this->username. ': ' . $this->password. ' là admin ' . $this->isAdmin;  
    }  
}
```



Thực thi câu truy vấn Select (FETCH_CLASS)



```
//Tạo câu truy vấn và thực thi
$stmt = $conn->query('SELECT * From USER);
//Thiết lập kiểu dữ liệu trả về, chỉ định đưa dữ liệu vào object của class
User
$stmt->setFetchMode(PDO::FETCH_CLASS, 'User');
//Hiển thị kết quả
while($obj = $stmt->fetch()) {
    echo $obj->username.'\n';
    echo $obj->password.'\n';
    echo $obj->isAdmin.'\n';
    Hoặc echo $obj->display().'\n';
}
```


Thực thi câu truy vấn Select (FETCH_CLASS)



- Kết quả trả về là một object theo class được chỉ định.
- Để constructor của class được gọi trước khi các thuộc tính được gán giá trị, sử dụng thêm **PDO::FETCH_PROPS_LATE**
- Để truyền tham số cho constructor của class thông qua fetch(): đặt các tham số trong một array theo thứ tự tương ứng

```
$stmt->setFetchMode(PDO::FETCH_CLASS | PDO::FETCH_PROPS_LATE, 'User');
```

```
$stmt->setFetchMode(PDO::FETCH_CLASS, 'User', array('param1', 'param2', 'param3'));
```

Thực thi câu truy vấn Insert, Update, Delete



- Đối với các câu lệnh Insert, Update, Delete (không có dữ liệu trả về) và không cần truyền tham số cho câu lệnh thì sử dụng phương thức **exec()** của đối tượng PDO

- VD:

```
$conn = new PDO("mysql:host=$SERVER;dbname=qlsv",$USERNAME,$PASSWORD);  
// Thiết lập lỗi của PDO cho ngoại lệ  
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
//Câu truy vấn  
$sql = "DELETE FROM USER WHERE id = 6";  
//thực hiện truy vấn  
$conn->exec($sql);
```



Sử dụng Prepared statements

- Sử dụng phương thức **prepare** của đối tượng PDO để thiết lập câu truy vấn

```
$stmt = $conn->prepare("INSERT INTO User (username, password) VALUES (:username, :pass)");
```

- Sử dụng **Name Parameters** để tạo liên kết tham số truyền dữ liệu và phương thức **bindParam** để gán các biến giá trị vào các tham số.

```
$stmt->bindParam(':username', $username);
```

```
$stmt->bindParam(':pass', $pass);
```

Hoặc dùng mảng: `$params = array(':username' => 'test', ':password' => '1234')`

- Dùng phương thức **execute** để thực thi truy vấn

```
$stmt->execute();
```

Hoặc: `$stmt->execute($param);`

Sử dụng Prepared Statements (Insert)



```
// Chuẩn bị câu lệnh INSERT
$sql = "INSERT INTO Persons (first_name, last_name) VALUES (:firstname, :lastname)";
$stmt = $conn->prepare($sql)
// Liên kết biến với tham số trong câu lệnh đã chuẩn bị
$stmt->bindParam(':firstname', $first_name);
$stmt->bindParam(':lastname', $last_name);
// Thiết lập tham số lần 1
$first_name = "abc";
$last_name = "xyz";
// Thực hiện câu lệnh
$stmt->execute();
// Thiết lập tham số lần 2
$first_name = "efg";
$last_name = "vmk";
// Thực hiện câu lệnh
$stmt->execute();
```



Sử dụng Prepared Statements (Select)



```
//Tạo Câu truy vấn dạng Preapared
$stmt = $conn->prepare('SELECT * From USER where username = :name');
//Thiết lập kiểu dữ liệu trả về
$stmt->setFetchMode(PDO::FETCH_ASSOC);
// Liên kết biến với tham số trong câu lệnh đã chuẩn bị
$stmt->bindParam(':name', 'abc');
// Thực hiện câu lệnh
$stmt->execute();
//Hiển thị kết quả
while($row = $stmt->fetch()) {
    echo $row['username'] , '\n';
    echo $row['password'];
}
```



Ngắt kết nối dữ liệu



- Ngắt kết nối cơ sở dữ liệu

```
$conn = null;
```



So sánh MySQLi và PDO



	MySQLi	PDO
CSDL hỗ trợ	Chỉ MySQL	12 loại CSDL
API	OOP + Procedural	OOP
Kết nối	Dễ dàng	Dễ dàng
Named parameters	Không	Có
Prepared statements	Có	Có
Ánh xạ đối tượng	Có	Có
Stored procedures	Có	Có
Hiệu năng	Cao	Cao



Tiếng Việt trong PHP và MySQL





<?php

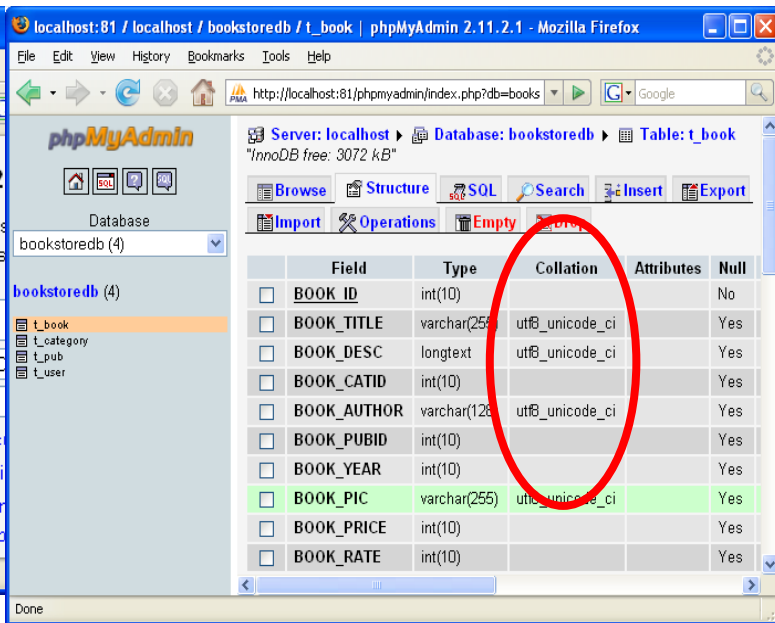
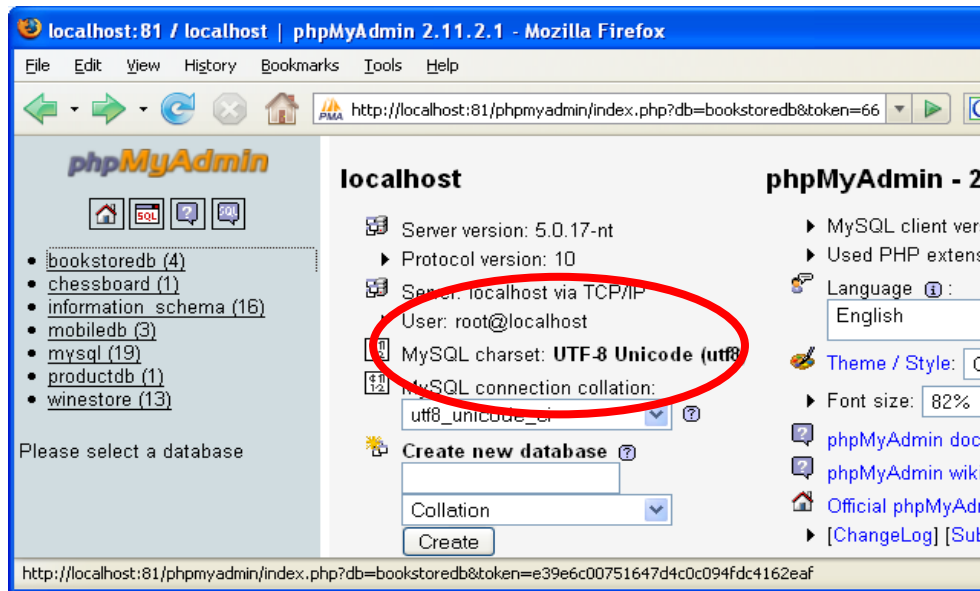
```
$server = "localhost";
$username = "root";
$password = "";
$dbName = "qlsv";
$conn = mysqli_connect($server, $username, $password, $dbName)
    Or die("<p> Không thể connect</p>");
$strSQL = "select * from User";
$result = mysqli_query($conn, $strSQL)
    Or die("<p>Không thể thực thi câu truy vấn.</p>");
echo "<table border =1 >";
echo "<tr><th>Username</th><th>Password</th></tr>";
while ($row = mysqli_fetch_row($result)){
    echo "<tr>";
    echo "<td>" . $row[1] . "</td>";
    echo "<td>" . $row[2] . "</td>";
    echo "</tr>";
};
echo "</table>";
//Đóng kết nối
mysqli_close($conn);
```

| user name | password |
|-----------------|----------|
| Mai Xuân Hồng | 0901 |
| Thị Bích Thị | 0902 |
| Nguyễn Hữu Trí | 0903 |
| Nguyễn Văn Phấn | 0904 |



MySQL & Font Unicode

- Tạo trong trình quản trị CSDL MySQL



MySQL & Font Unicode



- Dùng câu lệnh SQL để tạo
 - Thiết lập thuộc tính Database: Khi tạo CSDL MySQL thiết lập **CHARACTER SET utf8 COLLATE utf8_general_ci**;

```
CREATE DATABASE sinhvien DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

- Hoặc thiết lập các thuộc tính của bảng

```
CREATE TABLE SinhVien (  
    MaSV VARCHAR(8) NOT NULL,  
    HoTen VARCHAR(50),  
    NgaySinh DATE,  
    PRIMARY KEY (MaSV)  
) COLLATE utf8_general_ci;
```

MySQL & Font Unicode



- o Thiết lập các thuộc tính của field

```
CREATE TABLE SinhVien (  
    MaSV VARCHAR(8) NOT NULL,  
    HoTen VARCHAR(50) COLLATE utf8_general_ci NOT NULL,  
    NgaySinh DATE,  
    PRIMARY KEY (MaSV)  
);
```

Chú ý: Có thể sử dụng 1 trong 3 cách trên, tùy từng trường hợp lưu trữ; và có thể sử dụng COLLATE nào bắt đầu bằng utf8 đều được.(vd: *utf8_unicode_ci*, *utf8_bin*,...)

Thiết lập Unicode trong file .php



- Khai báo thẻ meta, charset=utf-8

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Trang quản lý thông tin SV</title>
</head>
```

- Đoạn code kết nối CSDL MySQL:

```
$conn = mysqli_connect($server, $user, $pass):
```

```
mysqli_set_charset($conn,"utf8");
```

```
Hoặc $conn->set_charset('utf8');
```

```
Hoặc PDO: $pdo->exec('SET NAMES "utf8"');
```

- Các thẻ form phải có thuộc tính method = “post”





Cảm ơn đã theo dõi

Hy vọng cùng nhau đi đến thành công.